



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/735,026	12/12/2003	James Blair Chapman	11235	2461
7590		04/19/2007	EXAMINER	
John D. Cowart Teradata Law IP, WHQ-4W NCR Corporation 1700 S. Patterson Blvd. Dayton, OH 45479-0001			PHAM, HUNG Q	
			ART UNIT	PAPER NUMBER
			2168	
SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE		DELIVERY MODE	
3 MONTHS	04/19/2007		PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/735,026	CHAPMAN, JAMES BLAIR	
	Examiner	Art Unit	
	HUNG Q. PHAM	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 14 February 2007.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-26 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-26 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

- 1) Certified copies of the priority documents have been received.
- 2) Certified copies of the priority documents have been received in Application No. _____.
- 3) Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____

5) Notice of Informal Patent Application

6) Other: _____

DETAILED ACTION

Response to Arguments

Claim Rejections - 35 USC § 101

Applicant' arguments with respect to the rejection under 35 U.S.C. § 101 have been fully considered but they are not persuasive.

- Claims 1-9 and 18-26 are directed to a method and system for redistributing data in a relational database management system. This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, if the conditional statement is not satisfied, the claimed subject matter provides for *executing a many-rows method to redistribute the one or more rows*¹. This produced result, e.g., *executing a many-rows method*, remains in the abstract and, thus, fails to achieve the required status of having real world value.

¹ MPEP 2111.04:

Claim scope is not limited by claim language that suggests or makes optional but does not require steps to be performed, or by claim language that does not limit a claim to a particular structure. However, examples of claim language, although not exhaustive, that may raise a question as to the limiting effect of the language in a claim are:

- (A) " adapted to " or "adapted for " clauses;
- (B) " wherein " clauses; and
- (C) " whereby " clauses.

The determination of whether each of these clauses is a limitation in a claim depends on the specific facts of the case. In *Hoffer v. Microsoft Corp.*, 405 F.3d 1326, 1329, 74 USPQ2d 1481, 1483 (Fed. Cir. 2005), the court held that when a "whereby" clause states a condition that is material to patentability, it cannot be ignored in order to change the substance of the invention." Id. However, the court noted (quoting *Minton v. Nat'l Ass'n of Securities Dealers, Inc.*, 336 F.3d 1373, 1381, 67 USPQ2d 1614, 1620 (Fed. Cir. 2003)) that a "whereby clause in a method claim is not given weight when it simply expresses the intended result of a process step positively recited." Id.<

Art Unit: 2168

- Claims 10 –17 are directed to a method for redistributing data in a relational database management system. This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, the claimed subject matter provides for *executing a few-rows redistribution method* or *executing a many-rows redistribution method*. This produced result, e.g., *executing a few-rows redistribution method* or *executing a many-rows redistribution method*, remains in the abstract and, thus, fails to achieve the required status of having real world value.

In light of the foregoing arguments, the 35 U.S.C. § 101 is hereby sustained.

Claim Rejections - 35 USC § 112

The rejection under 35 U.S.C. § 112, second paragraph, has been withdrawn in view of the amendment of claims 1, 11, 18 and 26.

Claim Rejections - 35 USC § 102

- Applicant's arguments with respect to the rejection under 35 U.S.C. § 102 have been fully considered but they are not persuasive.

As argued by applicant:

(1) At Page 15 Lines 3-19

... Ross does not teach or suggest performing different actions depending on whether the allocated buffer is larger than the one or more rows to be redistributed, as required by claims 1, 11, 18, and

26... Ross describes flushing a buffer to a mass storage disk if available allocated memory for that buffer becomes full, Ross col. 6, lines 22-25, and flushing records from an input table to main memory, Ross col. 6, lines 34-38, but does not teach or suggest performing one process if the allocated buffer (i.e., the "available main memory" according to this interpretation of the office action) is larger than the one or more rows to be redistributed and otherwise performing another process, as required by claims 1, 11, 18, and 26.

If the office action is equating Ross's buffers with the allocated buffer required by claims 1, 11, 18, and 26, then Ross does not teach or suggest a many-rows method to redistribute the one or more rows as required by claims 1, 11, 18, and 26, which the office action equates with Ross's partitioning method for distributing records from tables into buffers, Ross, col. 5, line 60 - col. 6, line 45. If the office action is equating Ross's buffers with the recipient of rows distributed through the many-rows method, Ross's buffers cannot be both the allocated buffer required by claims 1, 11, 18, and 26 and the recipient of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows, as required by claims 1, 11, 18, and 26.

(2) From Page 15-Line 20 to Page 16-Line 10

If the office action is equating Ross's output tables with the recipients of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows, as required by claims 1, 11, 18, and 26, then Ross is missing the requirement that the one or more rows be redistributed through a many-rows method if the allocated buffer is not larger than the one or more rows, as required by amended claims 1, 11, 18, and 26. The office action equates the many-rows method with Ross's partitioning method. In the portion of Ross referred to in the office action as the partitioning method, data is distributed through Ross's buffers, which the office action (under this interpretation) equates with the allocated buffer required by the claims, to Ross's output tables, which the office action (under this interpretation) equates with the recipients of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows. This is counter to what is required by claims 1, 11, 18 and 26, which require that, when the allocated buffer is not larger than the one or more rows, the one or more rows should be redistributed through a many-rows method, without first storing the one or more rows in the allocated buffer.

(3) At Page 16 Lines 11-15 and Page 17 Lines 8-16:

Further, Ross does not teach or suggest a many-rows method to redistribute the one or rows, as required by claims 1, 11, 18, and 26. Ross's partitioning method is not a many-rows redistribution method as suggested in the office action. An example of a many-rows method to

distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application.

...
... Ross does not teach or hint at executing a few-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25. An example of a few-rows distribution method is described in the specification in Fig. 6 and accompanying text...

Further, Ross does not hint at executing a many-rows redistribution method to redistribute the one or more rows, as required by claims 10 and 25. Ross's partitioning method is not a many-rows redistribution method as suggested in the office action. An example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application.

Examiner respectfully disagrees.

(1) As recited in the claims, after the step of allocating a buffer (Lines 3-5), the method of redistributing data either processes the step of *storing one or more rows... communicating a message...* (Lines 10-12) or *executing a many-row method* (Line 14) based on the size of the records, e.g., *if the allocated buffer is larger than the one or more rows* (Lines 8-9). In short, based on the conditional statement of Lines 8-9, either *storing one or more rows... communicating a message...* or *executing a many-row method* is performed.

As disclosed by Ross, to join two or more tables together, the available main memory (RAM) is allocated into partitions called buffers and assigned conditions such that all the unique record fulfilling each buffer condition will fit into available main memory (Col. 5 Lines 1-5 and 9-11). The number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers (Col. 5 Lines 13-21). As suggested by Ross, in some cases, table of data will be very small relative to the amount of main memory. An example of a relatively small table is one that contains data of 30 students' names and grades. In this example, the entire table can be read

into a typical main memory and can be completely processed while contained in RAM (Col. 1 Lines 36-46). Thus,

If two relatively small tables are joined together, and if the total size of the records to be read is less than the size of the available memory, only one buffer is allocated and the RAM will not be partitioned. The step of *storing one or more rows... communicating a message...* is disclosed by Ross at Col. 6 Lines 13-15 and 30-45

Otherwise, if the total size of the records to be read is larger than the size of the available memory, the available main memory (RAM) is partitioned, and the purpose is *to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45. The technique of partitioning the main memory (RAM) is considered as *executing a many-rows method*.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *Ross's buffers cannot be both the allocated buffer required by claims 1, 11, 18, and 26 and the recipient of rows distributed from the allocated buffer through the many-rows method of redistributing the one or more rows, as required by claims 1, 11, 18, and 26*) are not recited in the rejected claim(s). As recited in the claims, after the step of allocating a buffer (Lines 3-5), the method of redistributing data either processes the step of *storing one or more rows... communicating a message...* (Lines 10-12) or *executing a many-row method* (Line 14) based on the size of the records, e.g., *if the allocated buffer is larger than the one or more rows* (Lines 8-9). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

(2) With respect to the applicant's argument that Ross *is missing the requirement that the one or more rows be redistributed through a many-rows method if the allocated buffer is not larger than the one or*

more rows, as required by amended claims 1, 11, 18, and 26. This argument has been fully addressed by the examiner as indicated above.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *what is required by claims 1, 11, 18 and 26, which require that, when the allocated buffer is not larger than the one or more rows, the one or more rows should be redistributed through a many-rows method, without first storing the one or more rows in the allocated buffer*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

(3) In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., *Ross does not teach or suggest a many-rows method to redistribute the one or rows... an example of a many-rows method to distribute the one or more rows is described in Fig. 5 and accompanying text of the instant application... Ross does not teach or hint at executing a few-rows redistribution method ... example of a few-rows distribution method is described in the specification in Fig. 6 and accompanying text*) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In light of the foregoing arguments, the 35 U.S.C. § 102 is hereby sustained.

- Applicant's arguments with respect to the rejection of the dependent claims under 35 U.S.C. § 102 and 103 at Pages 16-24 have been fully considered but they are not persuasive. The dependent claims are unpatentable for at least those reasons as discussed above.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-26 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1-9 and 18-26 are directed to a method and system for redistributing data in a relational database management system. This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, if the conditional statement is not satisfied, the claimed subject matter provides for *executing a many-rows method to redistribute the one or more rows*². This produced result, e.g., *executing a*

² See footnote 1

many-rows method, remains in the abstract and, thus, fails to achieve the required status of having real world value.

Claims 10 –17 are directed to a method for redistributing data in a relational database management system. This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring article/phenomenon) since it fails to produce a tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. More specifically, the claimed subject matter provides for *executing a few-rows redistribution method* or *executing a many-rows redistribution method*. This produced result, e.g., *executing a few-rows redistribution method* or *executing a many-rows redistribution method*, remains in the abstract and, thus, fails to achieve the required status of having real world value.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-4, 6-20 and 22-25 are rejected under 35 U.S.C. 102(b) as being anticipated by Ross [USP 5,666,525].

Regarding claims 1 and 18, Ross teaches a method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

if the allocated buffer is larger than the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

storing one or more rows of a database table in the allocated buffer (Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

*communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer*³ (Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);

³ The step of communicating will end the method based on the conditional statement.

*otherwise*⁴:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45, if the).

Regarding claims 1 and 18, Ross teaches a method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

if the allocated buffer is larger than the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

storing one or more rows of a database table in the allocated buffer (Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

⁴ Because the allocated buffer is smaller than the one or more row, the method will not perform the steps of storing and communicating and branch to the step of executing.

*communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer*⁵ (Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*).

Regarding claims 1 and 18, Ross teaches a method for redistributing data in a relational data base management system, comprising:

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

if the allocated buffer is larger than the one or more rows:

storing one or more rows of a database table in the allocated buffer;

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer;

*otherwise*⁶:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method* to

⁵ See footnote 3.

⁶ See footnote 4.

redistribute the one or more rows as disclosed from Col. 5 Line 60 to Col. 6 Line 45 because the allocated buffer is smaller than the one or more rows).

Regarding claims 10 and 25, Ross teaches a method and system for redistributing data in a relational data base management system, comprising:

invoking a program on one or more of a plurality of transmitting modules, the program capable of managing of one or more rows associated with one or more database tables (Col. 4 Line 64-Col. 5 Line 1);
if the program was invoked on a single transmitting module; executing a few-rows redistribution method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned and be considered as *single transmitting module*. As disclosed at Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table is written to the available memory*);

otherwise;

*executing a many-rows redistribution method to redistribute the one or more rows*⁷ (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 6).

⁷ As recited in the claim, either few-row redistribution method or many-rows redistribution method is performed based on the conditional statement is satisfied or not.

Regarding claims 2 and 12, Ross teaches all the claim subject matters as discussed above with respect to claims 1 and 11, Ross further discloses *the message comprises all of the one or more rows stored in the allocated buffer* (Col. 6 Lines 30-45).

Regarding claims 3, 13 and 19, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses *the transmitting processing module comprises one of a plurality of processing modules associated with a relational database system* (Col. 6 Lines 13-18).

Regarding claims 4, 14 and 20, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses *the message is communicated to each of a plurality of destination processing elements* (Col. 6 Lines 30-45).

Regarding claims 6, 16 and 22, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 10 and 18, Ross further discloses *the many-rows method comprises: communicating from one or more transmitting processing modules a first signal to a plurality of processing modules within a relational database system, the first signal operable to initiate a row receiver task on each of the processing modules; from one or more of the processing modules a ready-to-receive signal to the one or more transmitting processing modules; communicating from the one or more transmitting processing modules a second signal comprising the one or more rows associated with the database table; after communication of the last row associated with the database table, communicating from the one or more transmitting processing modules an end-of-data signal to each of the plurality of processing modules* (Col. 5 Line 60 to Col. 6 Line 45).

Regarding claims 7 and 23, Ross teaches all the claim subject matters as discussed above with respect to claims 1 and 18, Ross further discloses *invoking the program on a single transmitting processing module* (As disclosed by Ross at Col. 5 Lines 13-21, if the total size of the records is less than the size of the available memory, the memory will not be partitioned and be considered as *single transmitting module*).

Regarding claim 8, Ross teaches all the claim subject matters as discussed above with respect to claim 1, Ross further discloses the step of *receiving at each of a plurality of destination processing elements a substantially similar set of the one or more rows stored in the allocated buffer* (Col. 6 Lines 30-45).

Regarding claims 9, 15 and 24, Ross teaches all the claim subject matters as discussed above with respect to claims 1, 11 and 18, Ross further discloses the step of *determining a number of rows to store in the allocated buffer* (Col. 5 Lines 1-21).

Regarding claim 11, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the steps of
allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a write operation as *transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows

(As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

storing one or more rows of a database table in the allocated buffer (Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

Regarding claim 11, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the steps of:

allocating a buffer associated with a transmitting processing module (Col. 5 Lines 1-5 and Col. 6 Lines 13-18, available memory as *a buffer* is allocated, the available memory is associated with a

Art Unit: 2168

write operation as transmitting processing module), the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (Col. 6 Lines 22-29, the write operation having access to the flush operation as a program, the flush operation is *capable of managing a redistribution of one or more rows associated with one or more database tables*);

comparing the allocated buffer to a portion of the buffer to be occupied by the one or more rows (Col. 5 Lines 18-21);

if the allocated buffer is larger than the portion of the buffer to be occupied by the one or more rows :

storing one or more rows of a database table in the allocated buffer ;

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Because the available memory is smaller than the total size of record, the method go to otherwise condition);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

Regarding claim 17, Ross teaches all the claim subject matters as discussed above with respect to claim 10, Ross further discloses the step of *determining the number of transmitting modules on which the program was invoked* (Col. 5 Lines 13-34).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 5 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ross [USP 5,666,525].

Regarding claims 5 and 21, Ross teaches all the claim subject matters as discussed above with respect to claims 1 and 18, Ross does not explicitly teach claimed limitation *the allocated buffer is capable of storing no more than ten (10) rows*. However, as disclosed at Col. 5 Lines 10-12, available memory is that portion which is not allocated to another function or to another user in a multi-user computer system. Thus, during a particular processing time, the available memory is reduced and *capable of storing no more than ten (10) rows*. It would have been obvious for one of ordinary skill in the art at the time the invention was made to include this feature into the available memory in order to partition the available memory.

Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Leverenz [Oracle8i Concept] in view of Ross [USP 5,666,525].

Regarding claim 26, Leverenz teaches a database management system (Page 17, Oracle Server), comprising:

a massively parallel processing system comprising (Page 53, Overview of Parallel Execution):

one or more nodes (Page 53, Overview of Parallel Execution, Oracle System);
a plurality of processors, each of the one or more nodes providing access to one or more processors (Page 53, Overview of Parallel Execution, Oracle System provides access to a plurality of CPUs); and
a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes (Page 54, Operation That Can Be Parallelized);
a set of one or more database tables residing on the one or more nodes (Page 61, FIG. 26-5, EMP Table); and

one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module (As shown and disclosed at Page 61 FIG. 26-5 and Page 58, How Parallel Execution Servers Communicate, scanning operations as *one or more of the plurality of virtual processes that allocate a buffer associated with ORDER BY operations as transmitting processing module*),
the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (As shown and disclosed at Page 61 FIG. 26-5, ORDER BY operations having access to Parallel Execution Coordinator as a program. The Parallel Execution Coordinator redistribute the query results include one or more rows to the user). The missing of Leverenz is the claimed limitation *if the allocated buffer is larger than the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows.*

Ross teaches a method for redistributing data in a relational data base management system, comprising:

if the allocated buffer is larger than the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected

records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

storing one or more rows of a database table in the allocated buffer (Ross, Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Ross, Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

The technique of partitioning the available is a must for Leverenz system in order to manage the available memory.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Ross into the Leverenz in order to manage buffers during parallel processing.

Regarding claim 26, Leverenz teaches a database management system (Page 17, Oracle Server), comprising:

a massively parallel processing system comprising (Page 53, Overview of Parallel Execution);
one or more nodes (Page 53, Overview of Parallel Execution, Oracle System);
a plurality of processors, each of the one or more nodes providing access to one or more processors (Page 53, Overview of Parallel Execution, Oracle System provides access to a plurality of CPUs); and
a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes (Page 54, Operation That Can Be Parallelized);
a set of one or more database tables residing on the one or more nodes (Page 61, FIG. 26-5, EMP Table); and
one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module (As shown and disclosed at Page 61 FIG. 26-5 and Page 58, How Parallel Execution Servers Communicate, scanning operations as *one or more of the plurality of virtual processes that allocate a buffer associated with ORDER BY operations as transmitting processing module*),
the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables (As shown and disclosed at Page 61 FIG. 26-5, ORDER BY operations having access to Parallel Execution Coordinator as *a program*. The Parallel Execution Coordinator redistribute the query results include one or more rows to the user). The missing of Leverenz is the claimed limitation *if the allocated buffer is larger than the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows.*

Ross teaches a method for redistributing data in a relational data base management system, comprising:

if the allocated buffer is larger than the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, "the number of buffers is determined by the total number and size of unique selected records. If the total size of the records to be read from a second input table during a join of two tables is three times the size of the available memory, the memory will be partitioned into three buffers". In different words, if the total size of the records is less than the size of the available memory, the memory will not be partitioned):

storing one or more rows of a database table in the allocated buffer (Ross, Col. 6 Lines 13-15, because the total size of the records is less than the size of the available memory, the *one or more rows of a database table* is written to the available memory);

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Ross, Col. 6 Lines 30-45, the records from the available memory is flushed to disk as *message comprising at least some of the one or more rows stored in the allocated buffer* for storing in an output file. The storing operation is considered as *one or more destination processing modules*);

otherwise:

executing a many-rows method to redistribute the one or more rows (The step of communicating will end the process because the conditional statement is satisfied).

The technique of partitioning the available is a must for Leverenz system in order to manage the available memory.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Ross into the Leverenz in order to manage buffers during parallel processing.

Regarding claim 26, Leverenz teaches a database management system (Page 17, Oracle Server), comprising:

a massively parallel processing system comprising (Page 53, Overview of Parallel Execution):

one or more nodes (Page 53, Overview of Parallel Execution, Oracle System);

a plurality of processors, each of the one or more nodes providing access to one or more processors

(Page 53, Overview of Parallel Execution, Oracle System provides access to a plurality of CPUs); and

a plurality of virtual processes, each of the one or more processors providing access to one or more virtual processes (Page 54, Operation That Can Be Parallelized);

a set of one or more database tables residing on the one or more nodes (Page 61, FIG. 26-5, EMP Table); and

one or more of the plurality of virtual processes that allocate a buffer associated with a transmitting processing module (As shown and disclosed at Page 61 FIG. 26-5 and Page 58, How Parallel Execution Servers Communicate, scanning operations as *one or more of the plurality of virtual processes that allocate a buffer associated with ORDER BY operations as transmitting processing module*), *the transmitting processing module having access to a program, the program capable of managing a redistribution of one or more rows associated with one or more database tables* (As shown and disclosed at Page 61 FIG. 26-5, ORDER BY operations having access to Parallel Execution Coordinator as *a program*. The Parallel Execution Coordinator redistribute the query results include one or more rows to the user). The missing of Leverenz is the processes of *if the allocated buffer is larger than the one or more rows: store one or more rows associated with a database table in the allocated buffer; communicate a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer; otherwise: execute a many-rows method to redistribute the one or more rows*.

Ross teaches a method for redistributing data in a relational data base management system, comprising:

if the allocated buffer is larger than the one or more rows :

storing one or more rows of a database table in the allocated buffer ;

communicating a message to one or more destination processing modules, the message comprising at least some of the one or more rows stored in the allocated buffer (Because the available memory is smaller than the total size of record, the method go to otherwise condition);

otherwise:

executing a many-rows method to redistribute the one or more rows (As disclosed by Ross at Col. 5 Lines 13-21, the partitioning method is considered as *a many-rows method to redistribute the one or more rows* as disclosed from Col. 5 Line 60 to Col. 6 Line 45).

The technique of partitioning the available is a must for Leverenz system in order to manage the available memory.

It would have been obvious for one of ordinary skill in the art at the time the invention was made to combine the teaching of Ross into the Leverenz in order to manage buffers during parallel processing.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HUNG Q. PHAM whose telephone number is 571-272-4040. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, TIM T. VO can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

H. Pham
HUNG Q PHAM
Examiner
Art Unit 2168

April 5, 2007